KeyTAR: Practical Keystroke Timing Attacks and Input Reconstruction

Mufan Qiu*, Lihsuan Chuang*, Dohhyun Kim, Huaizhi Qu, Tianlong Chen, Andrew Kwong

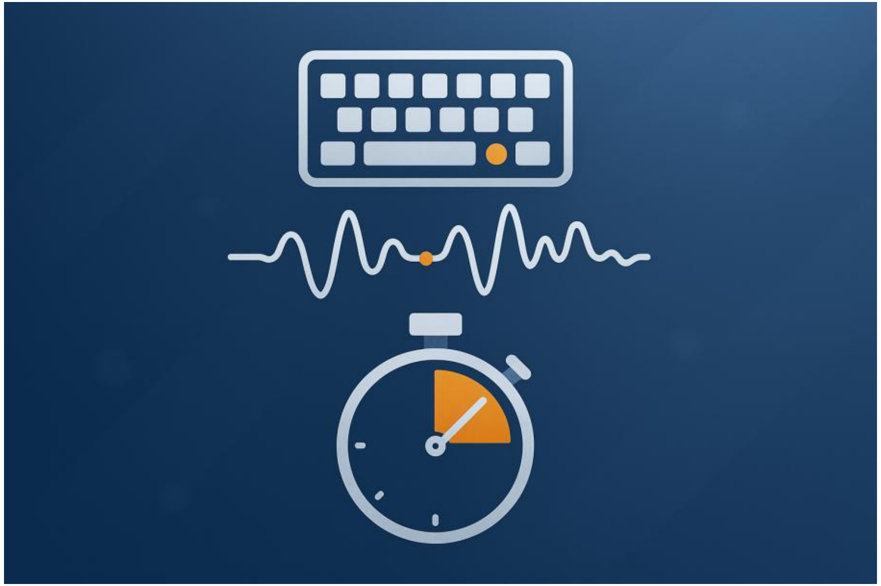
IEEE S&P 2026

Department of Computer Science

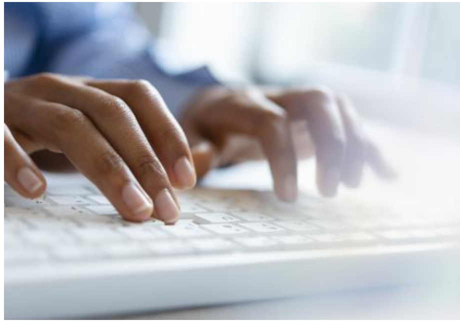
THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Overview

- Overview of Keystroke Timing Attack
- Keystroke Extraction
- Trace Collection
- Simulation Framework
- Input Reconstruction
- Results



Keystroke Timing Attack



Keystroke Extraction

Intervals: 74, 91, 108, 126, 143, 167, 182, 199, 214, 237, 255, 276, 289, 305, 328, 347, 362, 389, 421, 478

Lynn, got to the office OK.

Input Reconstruction

Keystroke Interrupt Handling

Key press

Interrupt Processing

1. Interrupt type
2. Key entered
3. Key destination

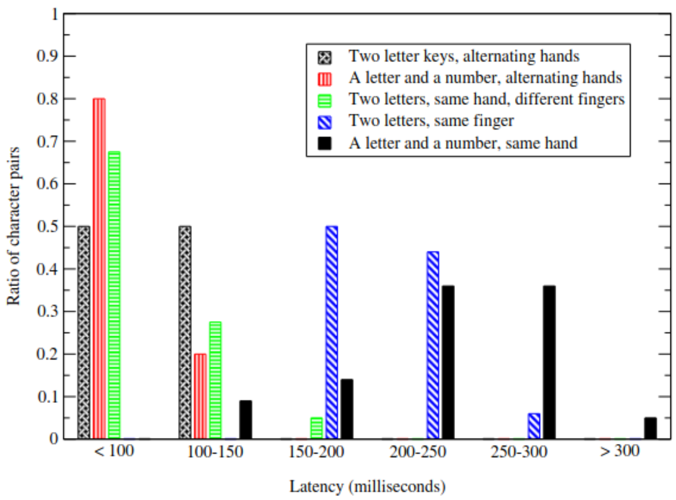
Handling Input

1. Screen Updates
2. Operations (Ctrl+C)
3. HTTP Request (Enter)
4. Network (ssh)

There are plenty of distinct executions unique to processing keystrokes

Why are these leaks dangerous?

Histogram of the latency of character pairs



Song, Dawn Xiaodong, David Wagner, and Xuqing Tian. "Timing analysis of keystrokes and timing attacks on {SSH}." *10th USENIX Security Symposium (USENIX Security 01)*. 2001.

Keystroke Extraction Techniques

SSH

Packet Arrival

SSH Keystroke Routines

Network

Network Traffic

Encoding

Interrupts

Direct Monitor

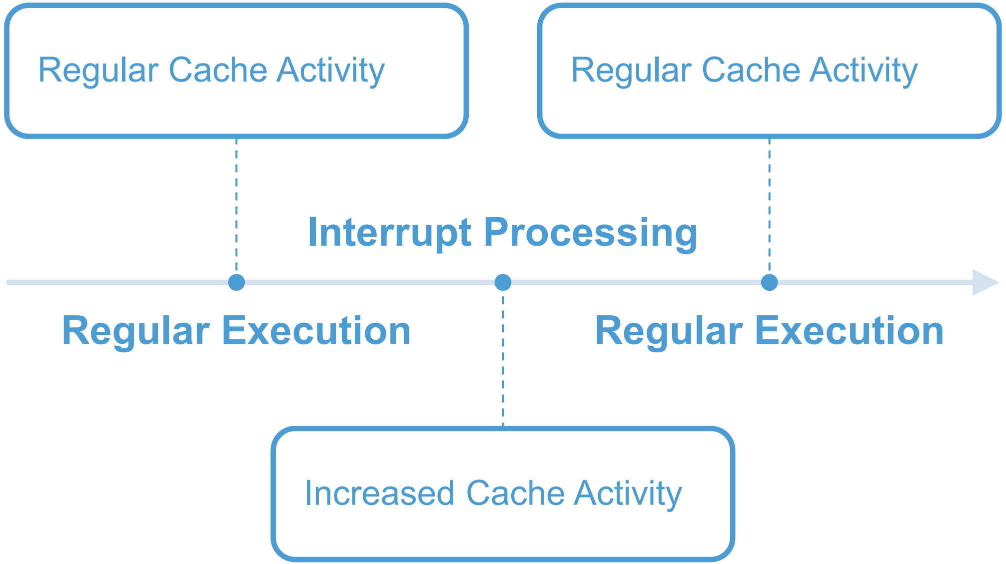
Indirect Monitor

Cache

Flush+Reload

Prime+Probe

Keystroke Extraction with Cache Attacks



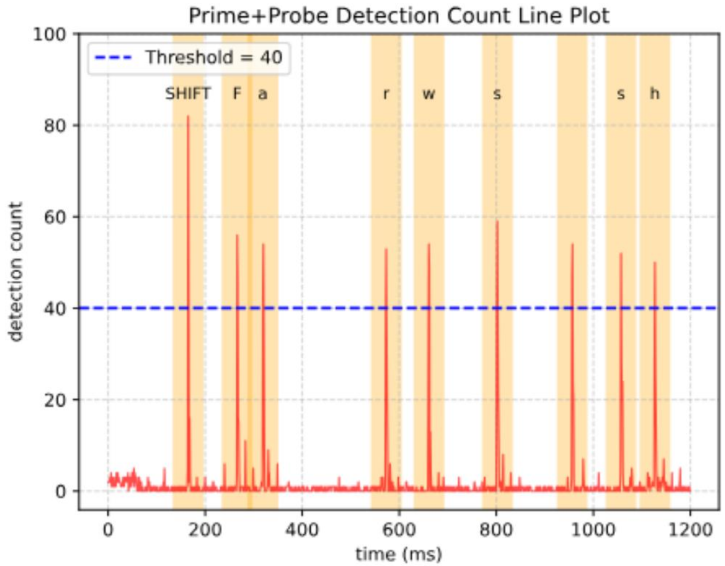
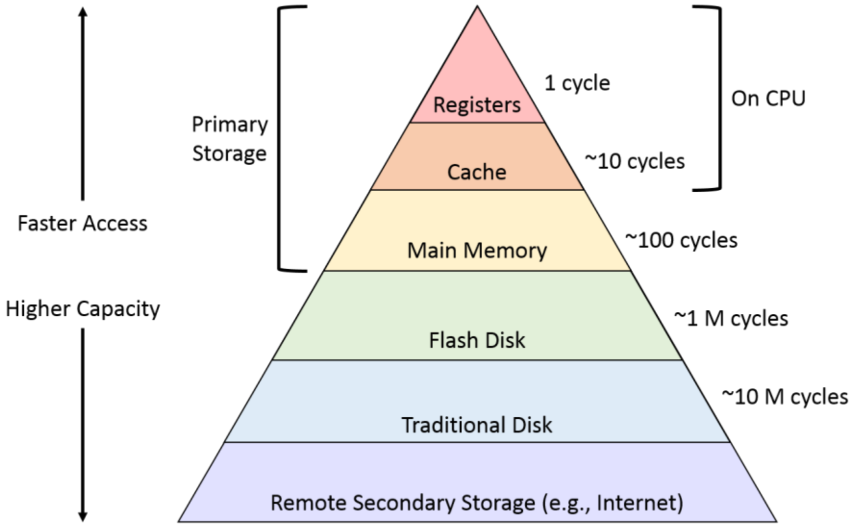
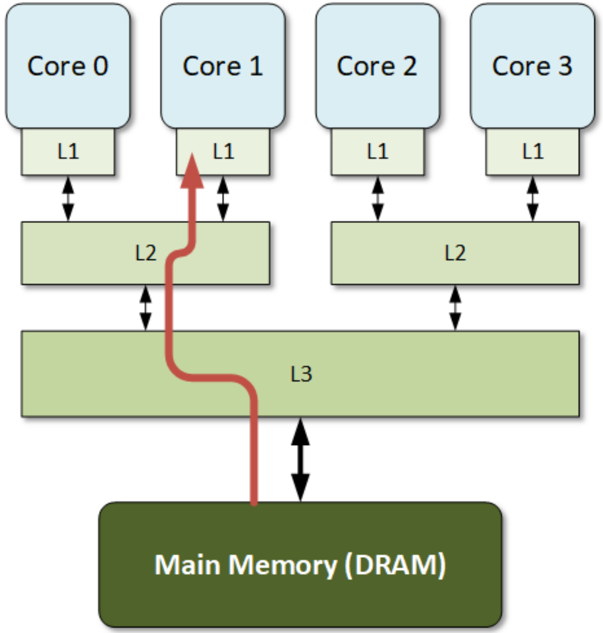


Figure 2. Keystroke Filtering from Aggregated Traces

Memory Hierarchy and The Cache

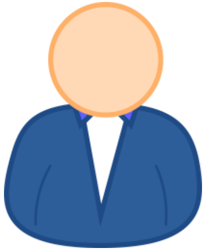


The Memory Hierarchy



Flush / Evict + Reload Extraction

Keystroke Handling Function (KHF)



Flush(KHF)

Type()

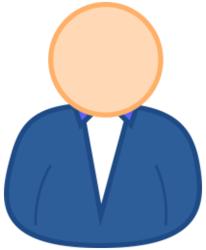
Wait(10000);

Cache

Reload(KHF)

Flush / Evict + Reload Extraction

Keystroke Handling Function (KHF)



Flush(KHF)

Type()

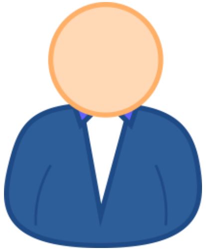
Wait(10000);

Cache

Reload(KHF)

Flush / Evict + Reload Extraction

Keystroke Handling Function (KHF)



Flush(KHF)

Type()

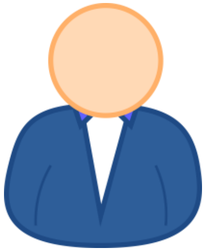
Wait(10000);

Cache

Reload(KHF)

Flush / Evict + Reload Extraction

Keystroke Handling Function (KHF)



Flush(KHF)

SLOW!!

Type()

Wait(10000);

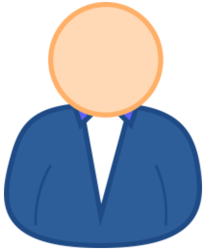
Cache

Reload(KHF)

Attacker infer that the victim did not type in the window.

Flush / Evict + Reload Extraction

Keystroke Handling Function (KHF)



Flush(KHF)

Wait(10000);

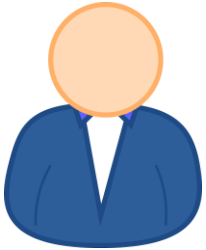
Reload(KHF)

Type()

Cache

Flush / Evict + Reload Extraction

Keystroke Handling Function (KHF)



Flush(KHF)

Type()

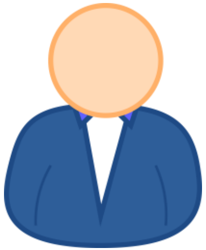
Wait(10000);

Cache

Reload(KHF)

Flush / Evict + Reload Extraction

Keystroke Handling Function (KHF)



Flush(KHF)

Type()

Wait(10000);

Cache

Reload(KHF)

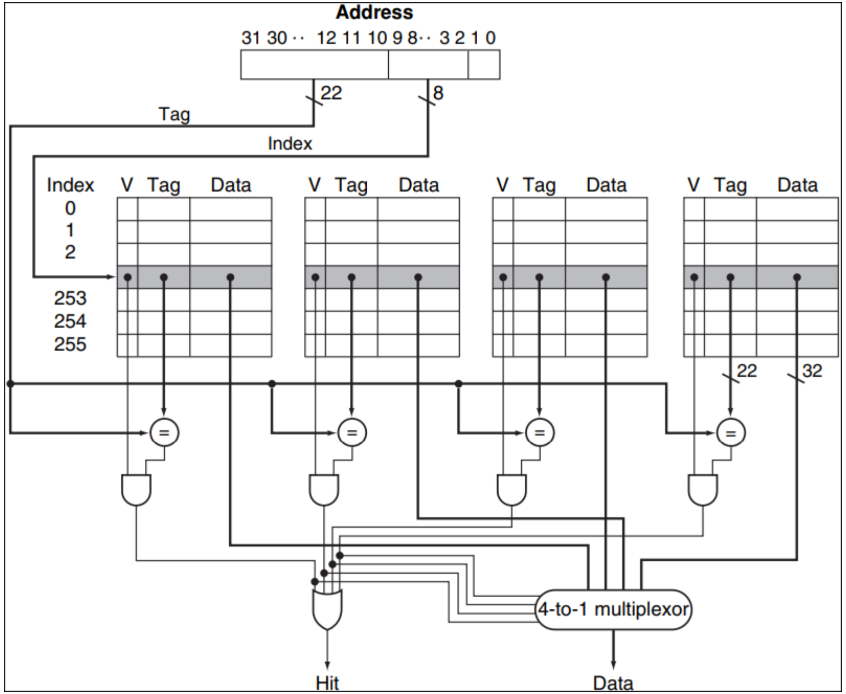
FAST!!

Attacker infer that the victim typed in the window.

Problems with Flush+Reload

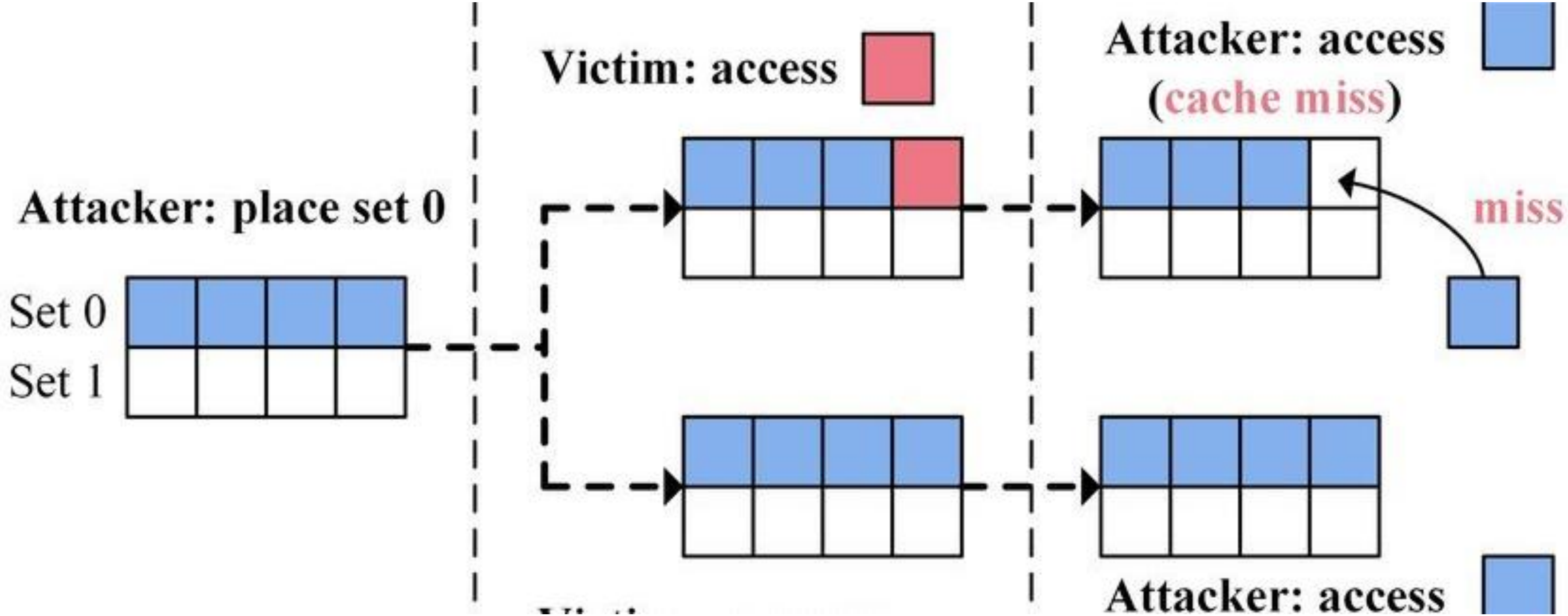
Memory	Requires shared memory
Target	Requires knowledge and access
Speed	Slow execution and blind spots
Generality	Unable to create a general exploit

Simplified Eviction Set Construction Example

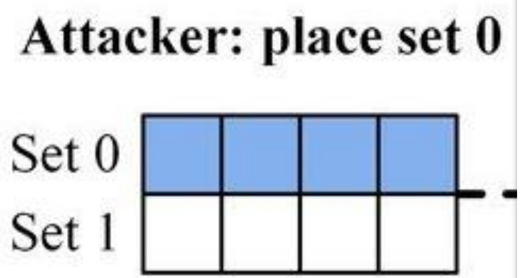


- 4-way set-associative cache
- 4-byte cache lines (2 bits)
- 256 sets (8 bits)
    - Usually computed by CACHE_SZ / WAYS / LINE_SZ
- Generate an eviction set with 4 lines with identical bits 2-9

Traditional Prime+Probe



Windowless Prime+Probe



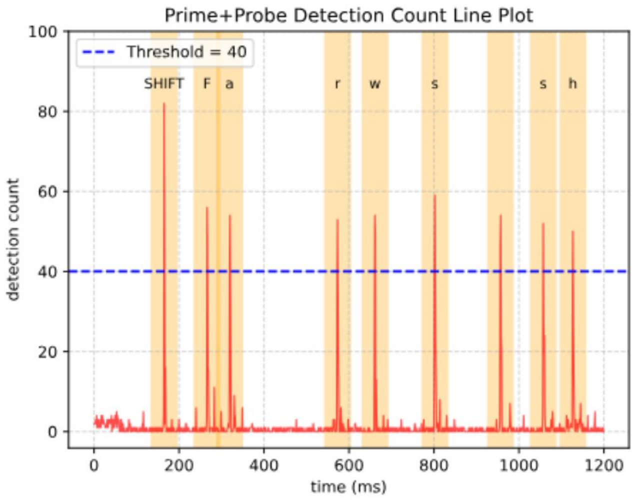
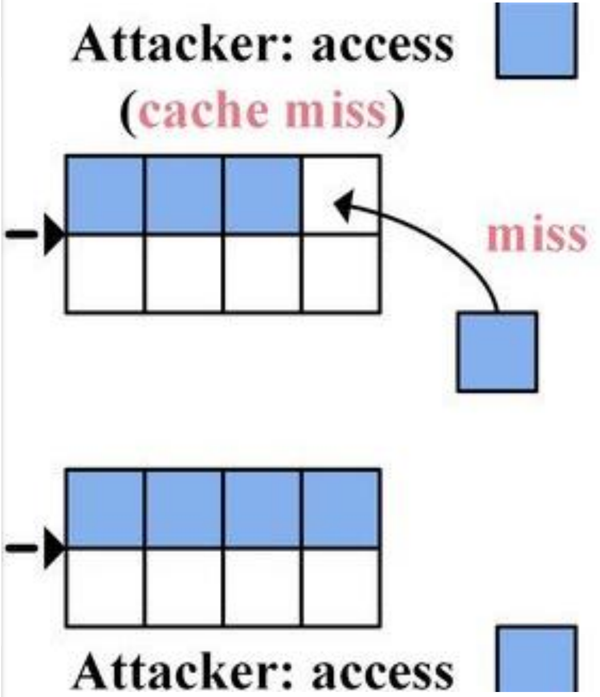


Figure 2. Keystroke Filtering from Aggregated Traces



Resolving Problems with Flush+Reload

Memory	Requires no shared memory
Target	Requires no knowledge or access
Speed	Fast execution and no blind spots
Generality	Monitor general activity of the cache

Threat Model

Native Extraction

- User-level attacker
- Execute arbitrary programs
- No software vulnerabilities

Web Extraction

- Server + Frontend Webpage
- Up-to-date browser with proper sandbox protection

Capability: Detect all keystrokes issued to the same device

Trace Collection

Dataset: Observations on Typing from 136 Million Keystrokes (Dhakal et. al. CHI 2018)

Simulate Keystroke Replays with IOCTL Interface

Trace Collection with Simultaneous Cache Attack

Simulation Framework

- **Generate side-channel traces for each typing sample**
- **Measuring Thread:**

Trace collection from interrupts

- **Simulating Thread:**

Replays typing samples

"test_id": "0-0-0",

"keystrokes": ["[SHIFT]", "t", "h", "e"],

"intervals_ms": [163, 254, 91, 143]

Figure: Example Simulation Input Data

Sequential Consistency (SC) Model

The result of any execution is the same as if the operations of all the processors were executed in some sequential order, and the operations of each individual processor appear in this sequence in the order specified by its program.

1. All instructions are executed in some order

2. Instructions within each program are executed in sequential order

Lamport, "How to Make a Multiprocessor Computer That Correctly Executes Multiprocess Programs," in IEEE Transactions on Computers, vol. C-28, no. 9, pp. 690-691, Sept. 1979, doi: 10.1109/TC.1979.1675439.

Sequential Consistency Execution Example

Process 1
A: X = 1
B: Y = 1
C: Print(Y)

Process 2
D: X = 0
E: Y = 0
F: Print(X)

X = 0, Y = 0	X = 0, Y = 1	X = 1, Y = 0	X = 1, Y = 1
A: X = 1	A: X = 1	A: X = 1	A: X = 1
B: Y = 1	D: X = 0	D: X = 0	B: Y = 1
D: X = 0	E: Y = 0	B: Y = 1	D: X = 0
E: Y = 0	B: Y = 1	E: Y = 0	E: Y = 0
C: Print(X)	C: Print(X)	C: Print(X)	C: Print(X)
F: Print(Y)	F: Print(Y)	F: Print(Y)	F: Print(Y)

Synchronization with Sequential Consistency

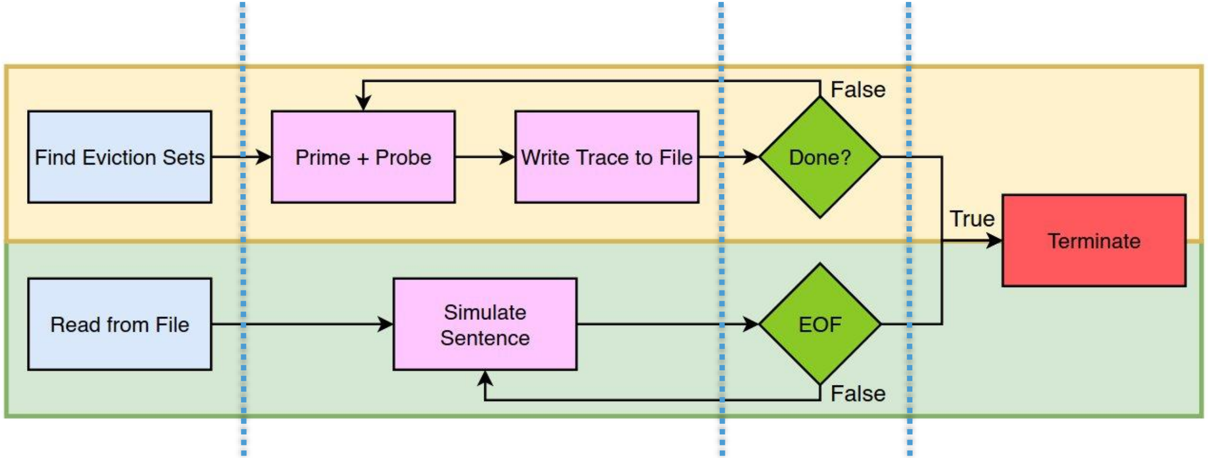
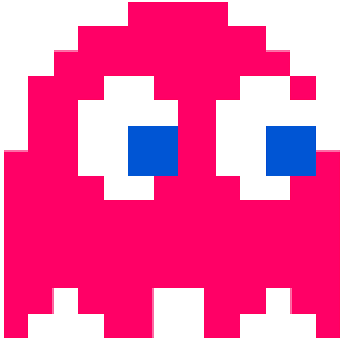


Figure 3. Simplified Keystroke Simulation System Workflow

- Concurrent attack and simulation
- Sync variables
  - EVSET_RDY
  - RD_FILE_DONE
  - IS_EOF
  - PP_RDY
- Shared Memory

Work in Progress: Attacking M2 on AVP

- Cache flush instructions
- Low-level memory interfaces
- Fine-Grained Timer
- Unique cache architectures
- Special optimizations (LSDP/MDP, DMP, LAP, LVP)



Remote Keylogging with Large Language Model

- Background: GPT Keylogger
  - What was your prompt? A Remote Keylogging Attack on AI Assistants (2024, USENIX)

rate tokens. For example, consider the text "*Oh no! I'm sorry to hear that. Try applying some cream.*" The tokenizer of GPT-3.5 and 4 would tokenize it as

Oh no! I'm sorry to hear that. Try applying some cream.

and the tokenizer of LLAMA-1 and 2 would tokenize it as

Oh no! I'm sorry to hear that. Try applying some cream.

LLM_B Training Prompt

Translate the Special Tokens to English, given the context.
Context: I need more details about your rash.
Special Tokens: _5 _3 _3 _1 _4 _5 _5 _3 _5 _5 _1

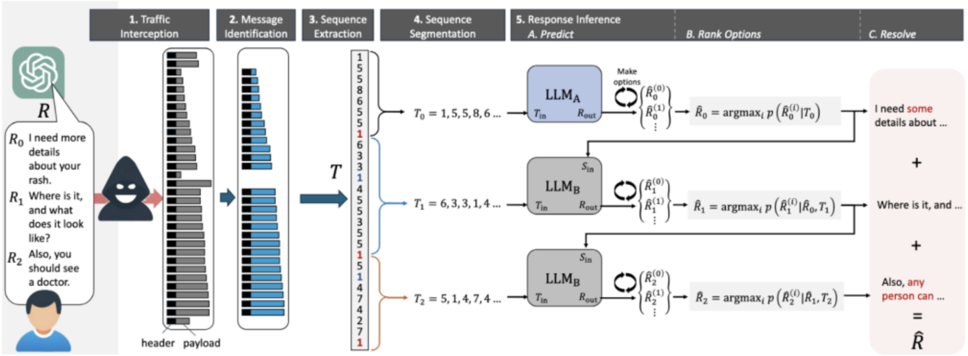


Figure 2: An overview of the attack framework: (1) Encrypted traffic is intercepted and then (2) the start of the response is identified. Then (3) the token-length sequence T is extracted and (4) a heuristic is used to partition T into ordered segments $(T_0, T_1, \ldots)$. Finally, (5) each segment is used to infer the text of the response. This is done by (A) using two specialized LLMs to predict each segment sequentially based on prior outputs, (B) generating multiple options for each segment and selecting the best (most confident) result, and (C) resolving the predicted response $\hat{R}$ by concatenating the best segments together.

Reconstruction on Clean Time Intervals

- Dataset:
  - Observations on Typing from 136 Million Keystrokes
- Method:
  - Machine translation task
- Metrics:
  - Treat edit distance < 0.1 as successful reconstruction
  - Dataset split
    - Within/Across Participants
    - Within/Across Sentences

Model	Top-1 Recon. Acc. ↑	Top-5 Recon. Acc. ↑
KREEP [25]	0.10%	0.20%
T5 (Ours) [50]	16.84%	33.53%
OLMo 1B (Ours)	**21.09%**	**34.92%**

LLM_A Training Prompt

User: Translate the Time intervals to Keystrokes.
Time intervals: 516 222 165 294 141 159 144 162 75 123 81 639 105 87 774 84 90 183 498 111 102 93 399 78 645 144 459
Assistant: Lynn, got to the office OK.

Representative Reconstruction Examples

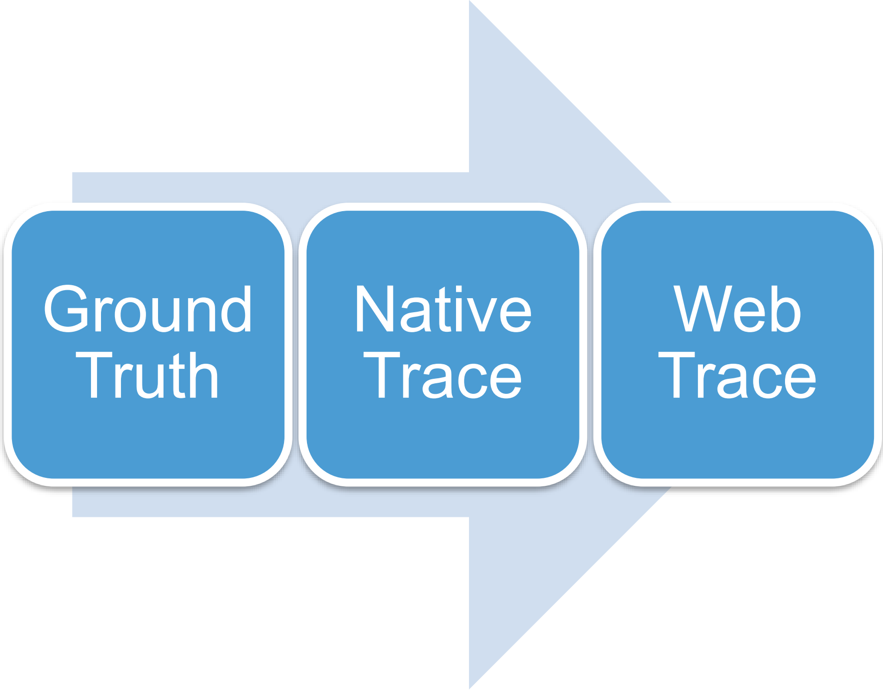
Edit distance ≈ 0.06
Input: Hope that all is well in Denver.
Prediction: Hope that all is well in Denver

Edit distance ≈ 0.07
Input: Crestone won't have final measurement until this week.
Prediction: crestone won't have final measurement until next week.

Edit distance ≈ 0.08
Input: Taka has to be completed.
Prediction: Task has to be completed.

Edit distance ≈ 0.17
Input: Let Gary Smith know if you want him.
Prediction: Let Gary Smith know today if you want him.

Reconstruction with Curriculum Learning



- Curriculum Learning: ML with tasks of increasing difficulty
- Noise determines the difficulty of input reconstruction

Reconstruction on Cache Time Intervals

- Dataset:
  - Observations on Typing from 136 Million Keystrokes
  - **Replayed and extract time intervals from cache**
- Method:
  - Modeling as machine translation task
  - **Curriculum Learning**
- Metrics:
  - Treat edit distance < 0.1 as successful reconstruction
  - Dataset split
    - Within/Across Participants
    - Within/Across Sentences

TABLE 3. ABLATION STUDY: TOP-5 RECONSTRUCTION ACCURACY (%) ON CACHE-EXTRACTED DATA

Training Strategy	APAS	APWS	WPAS	WPWS
Ground Truth Only	11.40%	25.05%	11.01%	25.39%
Cache Only	3.71%	19.41%	4.35%	19.14%
Curriculum Learning (Ours)	**16.94%**	**42.92%**	**20.21%**	**41.89%**

TABLE 2. RECONSTRUCTION PERFORMANCE OF THE CURRICULUM LEARNING MODEL ON CACHE-EXTRACTED TIME INTERVALS

Setting	Top-1 Recon. Acc. ↑	Top-5 Recon. Acc. ↑	Top-1 Mean Edit Dist. ↓	Top-5 Mean Edit Dist. ↓
APAS	8.84%	16.94%	0.7635	0.6251
APWS	26.78%	42.92%	0.6070	0.4294
WPAS	9.25%	20.21%	0.7437	0.5928
WPWS	27.15%	41.89%	0.6077	0.4382

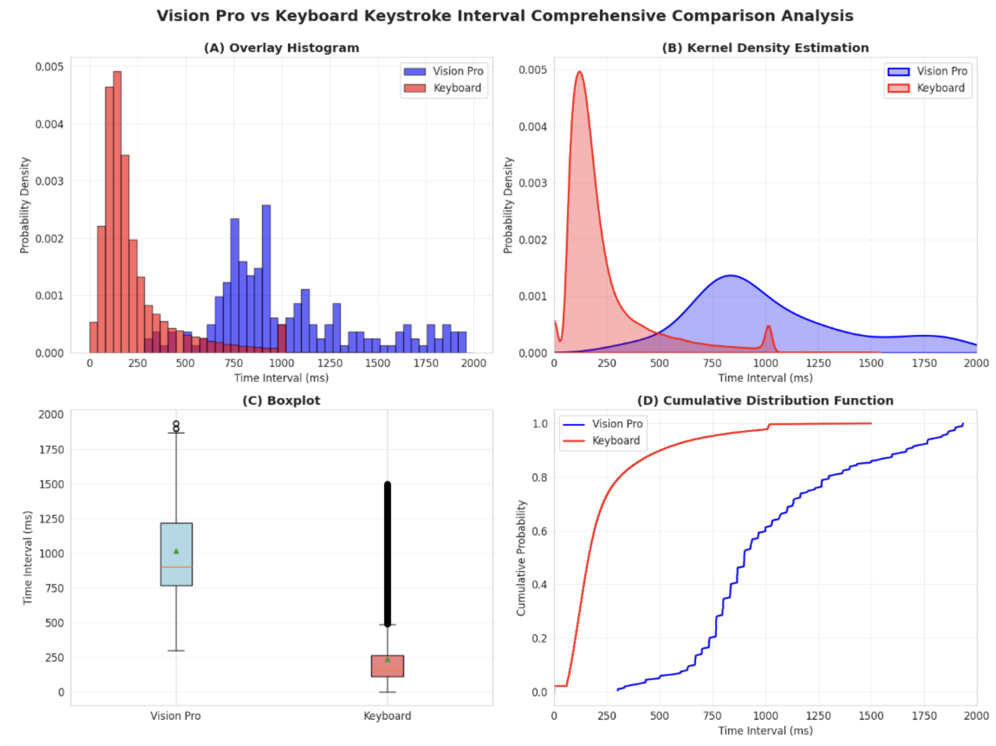
Extend to Vision Pro Input Traces

- Challenge:
  - Dataset is much smaller
  - Distribution shift

```
Vision Pro keystroke interval data points: 229
Keyboard keystroke interval data points: 479107

Vision Pro statistics:
  Mean: 1227.63 ms
  Median: 965.00 ms
  Std Dev: 687.39 ms
  Min: 301.00 ms
  Max: 5205.00 ms

Keyboard statistics:
  Mean: 235.24 ms
  Median: 162.00 ms
  Std Dev: 211.77 ms
  Min: 0.00 ms
  Max: 1500.00 ms
```



Extend to Vision Pro Input Traces

- Reasoning with question:
  - Step 1: Har Constraint – Keystroke Counting
  - Step 2:
    - Soft Constraint – Rhythm and Timing Analysis
    - Work Boundaries (Pauses)
    - Complexity & Speed
  - Step 3: Final Selection

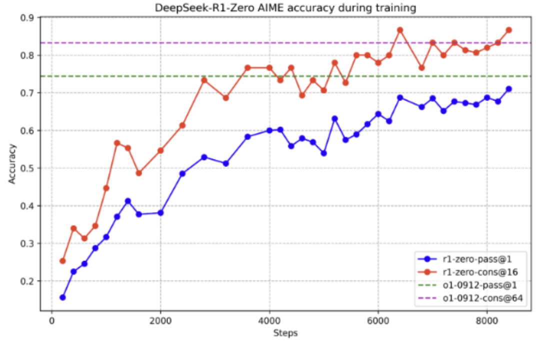


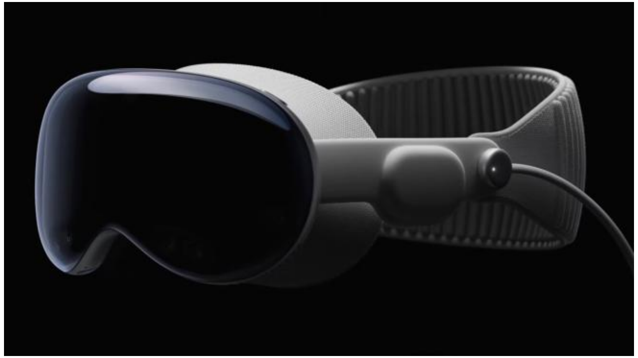
Figure 2 | AIME accuracy of DeepSeek-R1-Zero during training. For each question, we sample 16 responses and calculate the overall average accuracy to ensure a stable evaluation.

```
Step 2: Soft Constraint – Rhythm and Timing Analysis
1. Word Boundaries (Pauses):
    – Typists often pause slightly longer between words (before hitting Space) or at the start of a new word.
    – Look at the sequence of intervals. Are there distinct "spikes" or larger values (e.g., > 200–300ms)?
    – Count the number of significant pauses. Does this count roughly match the number of words (spaces) in the option?
2. Complexity & Speed:
    – Short intervals (e.g., < 100ms) often correspond to easy bigrams (e.g., 'th', 'er', 'in') or alternating hands.
    – Long intervals might correspond to Shift key presses, difficult reaches, or punctuation.
    – Does the "texture" of the intervals match the complexity of the sentence? (e.g., a simple sentence should have
    smooth intervals; a complex one with symbols should be choppier).
```

Extend to Vision Pro Input Traces

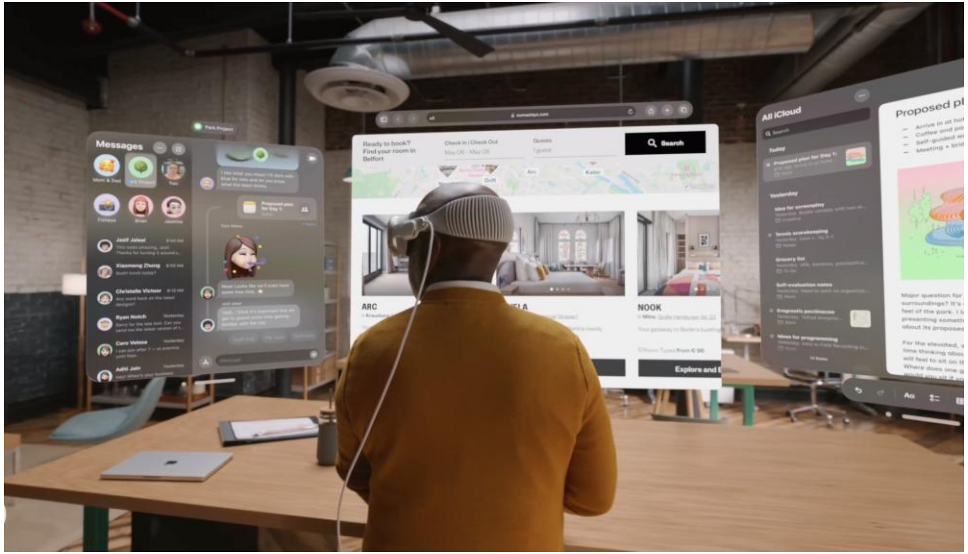
- Baseline: 1B LLM Model Finetuned on 1.3M time interval and sentence pairs
- Performance: 22% Top-1 reconstruction success rate on clean time interval and sentence pairs
- Experiment:
  - Synthesis 3000 output templates and retrain the baseline model with mixed outputs to preserve the language ability and enable the reasoning ability
    - 0.2M template outputs:
      - "<think> </think> Based on the inter-keystroke timings, the user appears to be typing <answer>...</answer>"
      - “<think> </think> These keystroke intervals likely correspond to the input <answer>...</answer>”
    - 10K Reasoning outputs generated by DeepSeek R1:
      - “<think> Follow step 1, I should…</think> User inputs <answer>...</answer>”
  - Performance: 24% successful reconstruction rate and model learns to follow the format
- Future Steps: reinforcement learning.

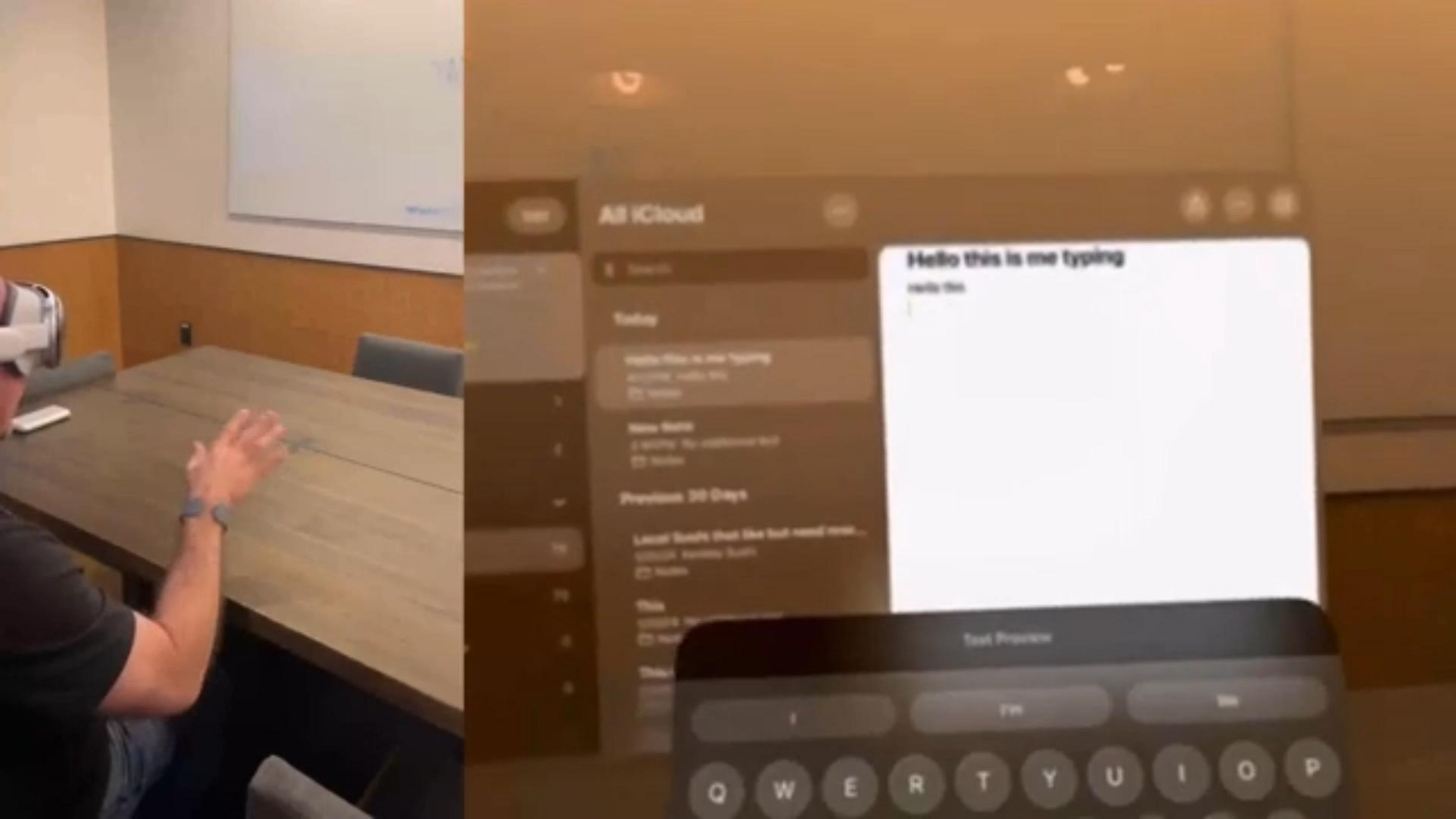
Background: Apple Vision Pro



Productivity

A workspace with infinite space.





All iCloud
Hello this is me typing
Today
Previous 30 Days
Text Preview
Q
W
E
R
T
Y
U
I
O
P

Extending to Apple Vision Pro: Pinch Typing

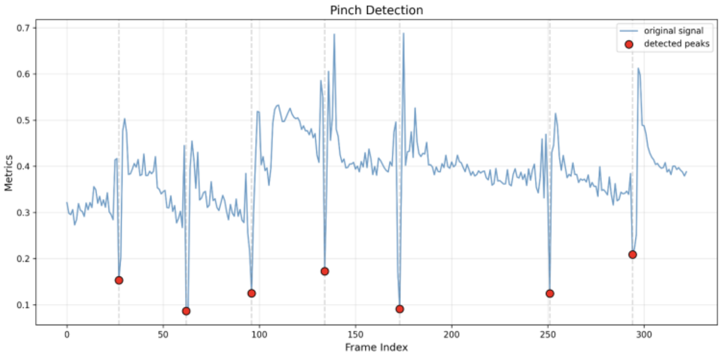
No existing traces available

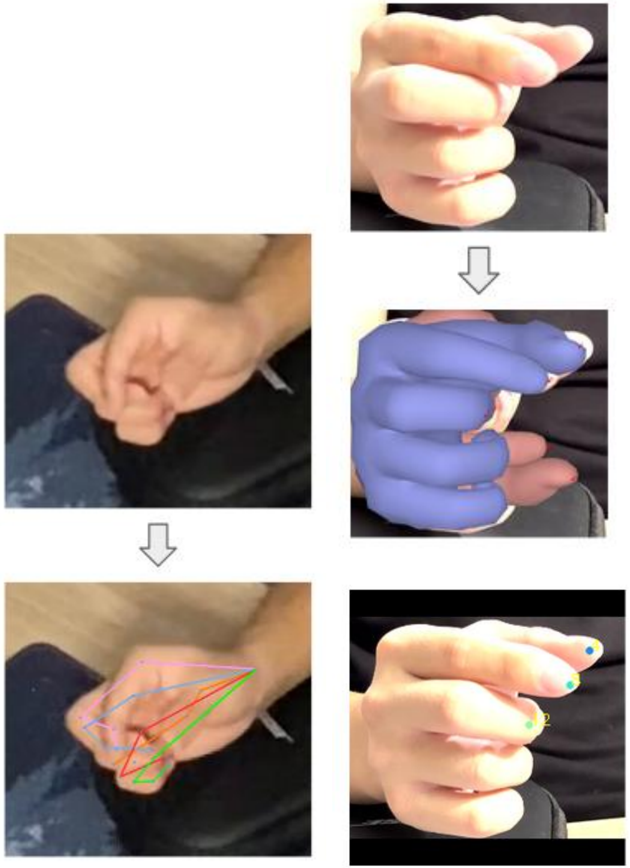
Replay framework does not exist

Goal: Generate a similar dataset for typing on the Apple Vision Pro

Background: Pose Estimation

- Pose estimation task is a well-established field with strong models
- Models can track each finger and map them onto a 3D Cartesian plane





Motivation

- Head-mounted devices (i.e. Apple Vision Pro) have been growing in popularity for productive use

- Preliminary studies show significant difference in regular and AVP keystroke timings due to its unique input method

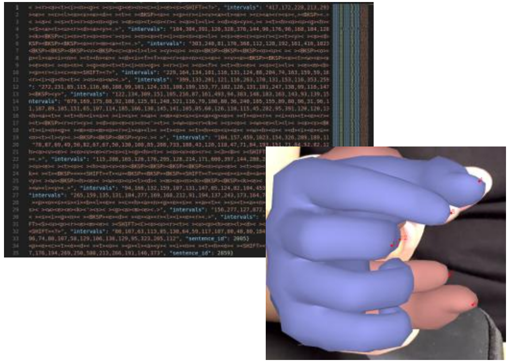
- Preliminary studies show strong results in detecting typing gesture

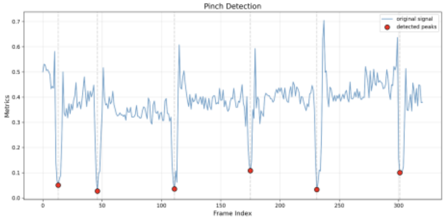
KeyTAR2.0 Workflow

Dataset Creation/ Model Training

Attack via P-P/Vision for Keystroke Timings

Typed Content Inference with LLM

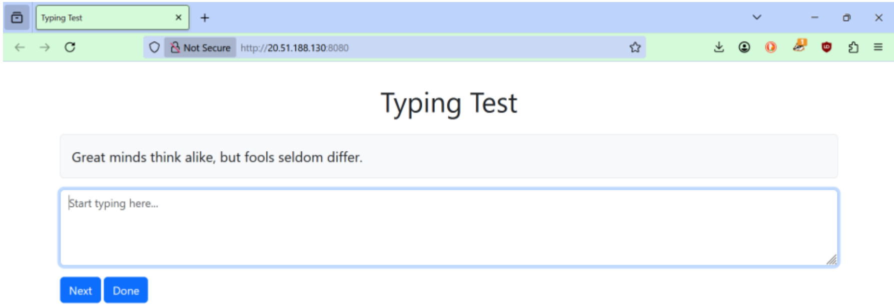




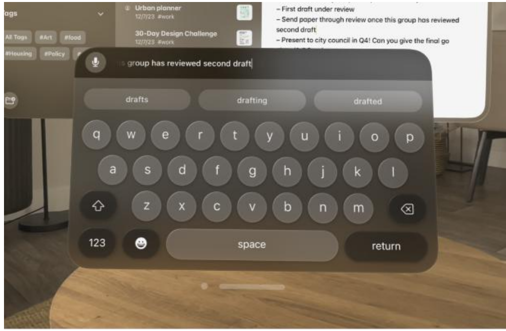
LLM_A Training Prompt

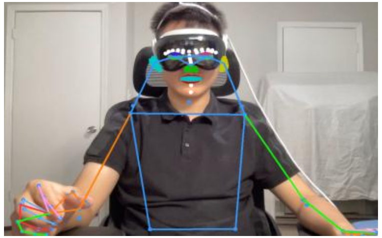
User: Translate the Time intervals to Keystrokes.
Time intervals: 516 222 165 294 141 159 144 162 75 123 81 639 105 87 774 84 90 183 498 111 102 93 399 78 645 144 459
Assistant: Lynn, got to the office OK.

Methodology: Collection

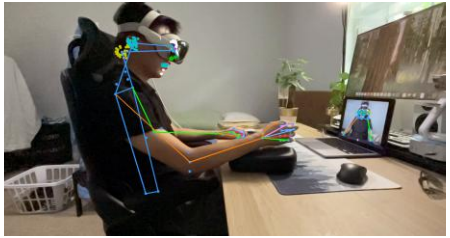


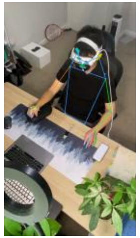
- 4 Perspectives
- Typing test on website
- Changed prompts to lowercase, no special characters
- Simultaneously run prime+probe to collect noisy traces




Front


Hand

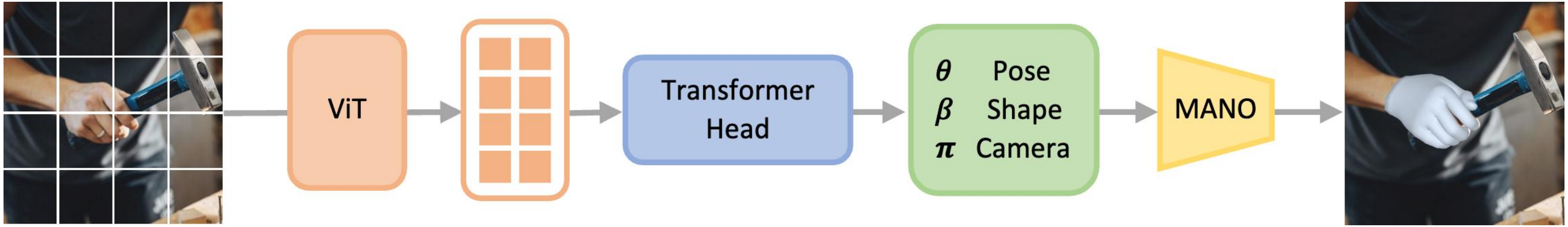

Side


Top

Methodology: Collection

- 4 * BU505MCF
  - 2,448 x 2,048, 75fps, USB3.1
- Sync four perspective through integrated controller (< 1ms)
- Sync videos with ground truth by server time request (< 30ms)

Methodology: Pinch Detection



1. Pose estimation model is used to locate and crop out the hand region
2. Use a ViT backbone to extract the visual feature from the hand image
3. Transformer-based decoder is applied to predict the parameters
4. The model is trained on a mixture of multiple datasets

Methodology: Pinch Detection

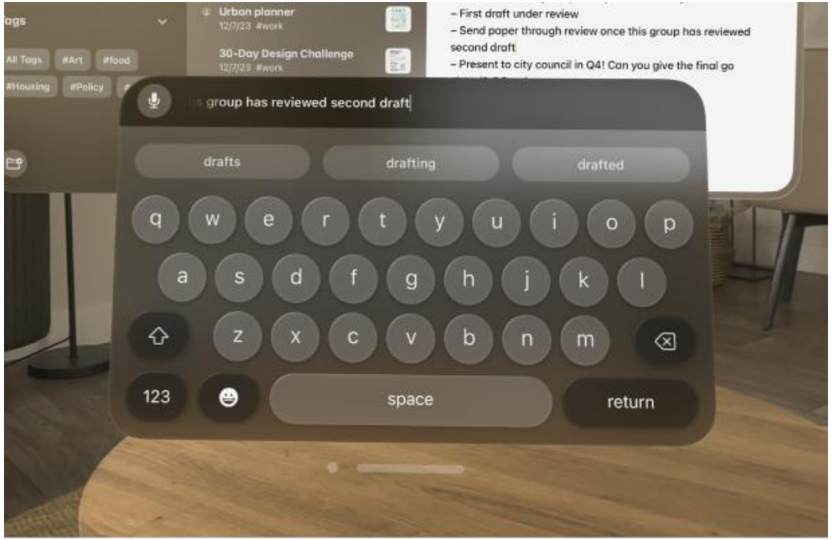
	170001	i_am_a_student	it_is_a_good_day	panzer	uzumymw
GT	6	14	16	6	7
front	6	14	16	6	7
hand	6	14	16	6	7
side	6	7	12	4	5
top	6	14	16	6	7

- The detection pipeline could accurately capture all the pinches in all the angles
- Except the very challenging side view, where the thumb is usually invisible

Methodology: Data Collection

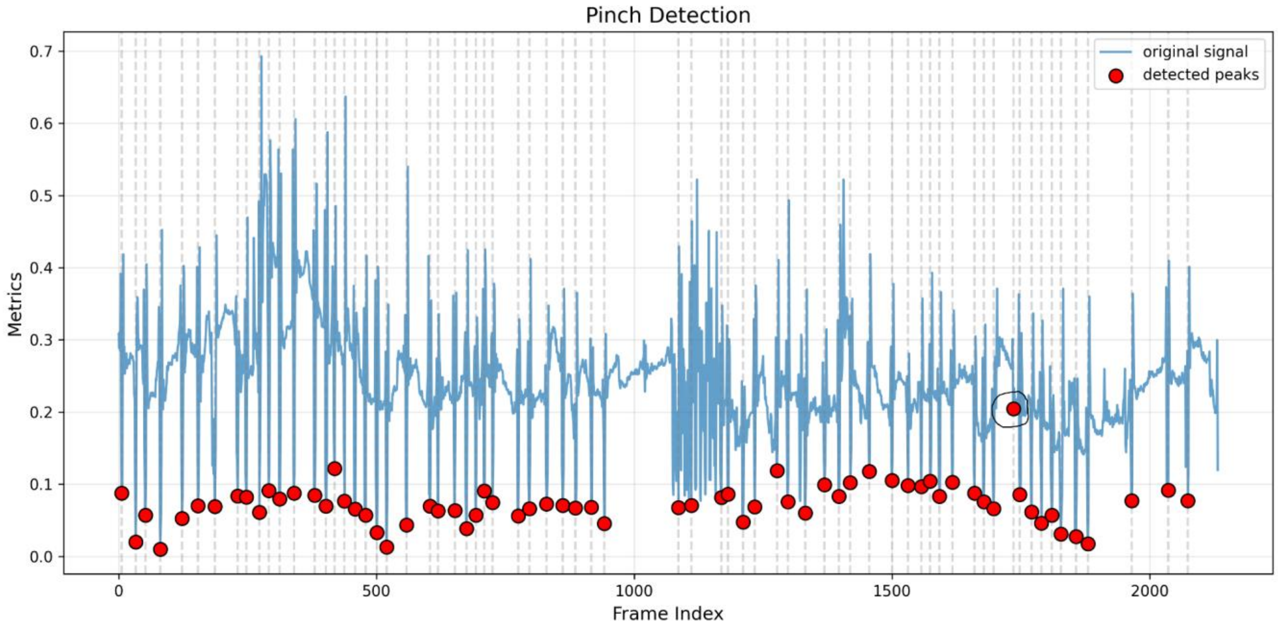
We first work on long sentences, but this is poses challenges:

- High incorrect typing rate due to the inaccurate eye tracking
- Mismatch between actual pinches and collected ground truth pinches




```
predicted time interval:1480.3987816307406, gt time interval:1000
predicted time interval:1760.4742268041236, gt time interval:366
predicted time interval:1240.334114339269, gt time interval:1534
predicted time interval:1040.2802249297094, gt time interval:938
predicted time interval:680.1832239925023, gt time interval:1062
predicted time interval:720.1940018744143, gt time interval:1433
predicted time interval:1000.2694470477976, gt time interval:1767
predicted time interval:1720.4634489222117, gt time interval:1599
predicted time interval:720.1940018744143, gt time interval:668
predicted time interval:760.2047797563263, gt time interval:735
predicted time interval:1560.420337394564, gt time interval:698
predicted time interval:480.1293345829428, gt time interval:1000
predicted time interval:920.2478912839738, gt time interval:1700
predicted time interval:760.2047797563263, gt time interval:733
predicted time interval:800.215557638238, gt time interval:767
predicted time interval:720.1940018744143, gt time interval:2065.000000000233
predicted time interval:1160.312558575445, gt time interval:867.9999999997672
predicted time interval:920.2478912839738, gt time interval:766
predicted time interval:3400.9161199625114, gt time interval:801
predicted time interval:2840.7652296157453, gt time interval:733
predicted time interval:1520.4095595126525, gt time interval:2066
未检测出pinch time如下:[1613755]
detect 69 peaks, index: [   6   33   52   81  123  154  187  231  248  273  291  312  340  380
  402  419  438  459  479  501  520  558  604  619  652  675  693  709
  725  775  797  830  861  886  917  942 1086 1111 1169 1182 1211 1234
 1277 1298 1332 1369 1397 1419 1456 1500 1531 1557 1574 1592 1617 1660
 1678 1697 1736 1748 1771 1790 1810 1828 1857 1880 1965 2036 2074]
```

Methodology: Data Collection



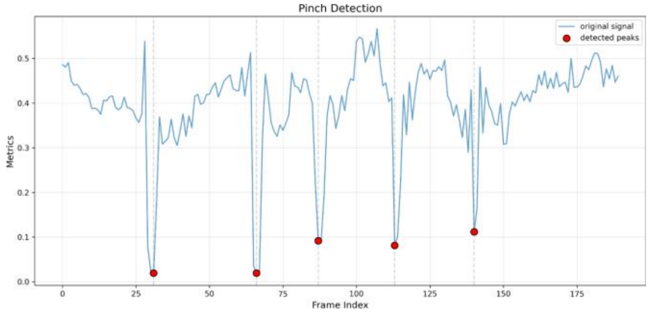
Methodology: Data Collection

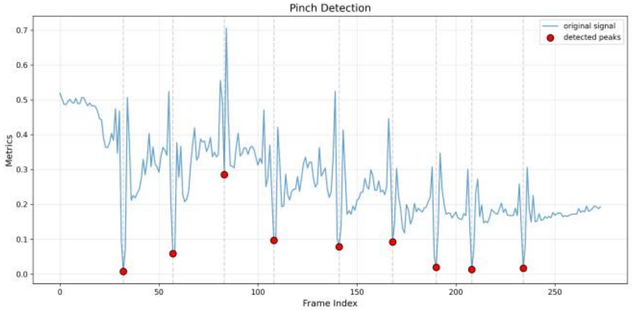
If we use simpler sentences and only considers keystrokes that can be captured on the Vision Pro keyboard

- Our method is pretty accurate in the time interval

```
/root/autodl-tmp/VR/data_new/IMG_7656.mp4
6378
predicted time interval:1168.7434554973822, gt time interval:1200
predicted time interval:701.2460732984293, gt time interval:698.9999999999709
predicted time interval:868.2094240837697, gt time interval:866
predicted time interval:901.6020942408378, gt time interval:867
detect 5 peaks, index: [ 31  66  87 113 140]
```

```
/root/autodl-tmp/VR/data_new/IMG_7655.mp4
9172
predicted time interval:833.8181818181819, gt time interval:835.0000000000291
predicted time interval:867.170909090909, gt time interval:835.9999999999709
predicted time interval:833.8181818181819, gt time interval:864
predicted time interval:1100.6399999999999, gt time interval:1100
predicted time interval:900.5236363636363, gt time interval:900
predicted time interval:733.76, gt time interval:734.0000000000291
predicted time interval:600.3490909090909, gt time interval:629.9999999999709
predicted time interval:867.170909090909, gt time interval:834.0000000000291
detect 9 peaks, index: [ 32  57  83 108 141 168 190 208 234]
```

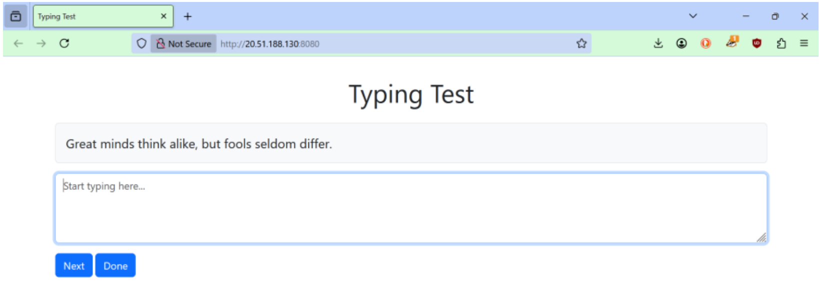




KEYTAR2.0: APPLE VISION PRO

Full Scale Data Collection

- ~15 Min
- Typing test on Apple Vision Pro
- Experienced vs Unexperienced typists
- Chance to play around and experience new tech!
- 2 Class bonus points!!!



Help us out!

THE UNIVERSITY
of NORTH CAROLINA
at CHAPEL HILL